Extractor Codes

Amnon Ta-Shma [*]
Computer Science Department
Tel-Aviv University
Israel 69978
amnon@post.tau.ac.il

David Zuckerman [†]
Department of Computer Science
University of Texas
Austin, TX 78712
diz@cs.utexas.edu

ABSTRACT

We define new error correcting codes based on extractors. We show that for certain choices of parameters these codes have better list decoding properties than are known for other codes, and are provably better than Reed-Solomon codes. We further show that codes with strong list decoding properties are equivalent to slice extractors, a variant of extractors. We give an application of extractor codes to extracting many hardcore bits from a one-way function, using few auxiliary random bits. Finally, we show that explicit slice extractors for certain other parameters would yield optimal bipartite Ramsey graphs.

1. INTRODUCTION

An error-correcting code is used to transmit information over noisy channels. If the received word differs from the transmitted word in not too many places, the closest codeword to the received word will equal the transmitted word. If such unique decoding is required, then error correcting codes cannot correct a number of errors which is more than half the minimum distance of the code. In particular, error correcting codes cannot give unique decoding for error rate greater than half.

List decoding allows recovery from a larger number of errors; in fact, for large alphabets the error rate can be close to 1. In this case, the list decoding problem requires that all codewords with modest agreement with the received word be output. If the number of errors is not overwhelmingly large, then the transmitted word will appear on this list.

List decoding may be possible under even harsher conditions. Imagine an extremely noisy channel, where for the ith alphabet symbol sent, the receiver only knows that the ith received symbol belongs to some $\mathcal{S}_i$ whose size is half the alphabet size. Then the appropriate list decoding problem is to find all codewords $w = (w_1, \ldots, w_T)$ such that $w_i \in \mathcal{S}_i$ for every i. An even harder problem is to to find all codewords $w = (w_1, \ldots, w_T)$ such that the fraction of $w_i \in \mathcal{S}_i$ is significantly more than $1/2$.

1.1 Some definitions

Let $\mathcal{C} \subseteq [\mathrm{M}]^{\mathrm{T}}$ be a code of size C.[1] We call the elements of $[\mathrm{T}] \times [\mathrm{M}]$ *points*. We can view a word $w \in [\mathrm{M}]^{\mathrm{T}}$ as the set of points $\{(i, w_i) \ : \ i \in [\mathrm{T}]\}$. The *agreement* between a word w and a set of points $\mathcal{S}$ is $\mathsf{Ag}(w, \mathcal{S}) = |\{(i, b_i) \in \mathcal{S} \ : \ b_i = w_i\}|$. The list decoding problem comes in two flavors, combinatorial and algorithmic. In both we are given a set $\mathcal{S}$ of points. The combinatorial version is to compute the *number* of codewords that have large agreement with $\mathcal{S}$, while the algorithmic version is to actually find all these codewords. Formally,

DEFINITION 1.1 (COMBINATORIAL VERSION). *Let* $\mathcal{C} \subseteq [\mathrm{M}]^{\mathrm{T}}$ *be a code and* $\mathcal{S} \subseteq [\mathrm{T}] \times [\mathrm{M}]$ *a set of points. The A-neighborhood of a set* $\mathcal{S}$ *is* $\Gamma_{\mathrm{A}}(\mathcal{S}) = \{w \in \mathcal{C} \ : \ \mathsf{Ag}(w, \mathcal{S}) \geq \mathrm{A}\}$. *The neighborhood size function is*

$$\mathsf{N}Sc(\mathrm{S}, \mathrm{A}) \;=\; \max_{\mathcal{S} : |\mathcal{S}| = \mathrm{S}} |\Gamma_{\mathrm{A}}(\mathcal{S})|$$

We say $\mathcal{C}$ *can be list decoded for parameters* $\mathrm{S} = \mathrm{S}(\mathrm{T}, \mathrm{M}), \mathrm{A} = \mathrm{A}(\mathrm{T}, \mathrm{M})$ *if* $\mathsf{N}Sc(\mathrm{S}, \mathrm{A}) = \mathrm{poly}(\mathrm{M}, \mathrm{T})$.

Of course, for the poly to make sense we need an infinite family of codes, which we implicity assume throughout the paper.

DEFINITION 1.2 (ALGORITHMIC VERSION). *We say* $\mathcal{C}$ *has an efficient list decoding for parameters* S, A *if there is a polynomial-time algorithm that given oracle access to the characteristic function of a set of points* $\mathcal{S}$ *of size* S, *outputs all codewords in* $\Gamma_{\mathrm{A}}(\mathcal{S})$ *in time* $\mathrm{poly}(\mathrm{M}, \mathrm{T})$. *We say* $\mathcal{C}$ *has efficient encoding if there is a polynomial-time algorithm that given* i *computes the* i*'th codeword of* $\mathcal{C}$ *in time* $\mathrm{poly}(\mathrm{M}, \mathrm{T})$.

Notice that we can represent a set $\mathcal{S} \subseteq [T] \times [M]$ with MT bits, so we could have replaced the above definition with one

[1]As usual, $[\mathrm{M}]$ denotes the set $\{1, 2, \ldots, \mathrm{M}\}$. We use unconventional letters because we avoid N, K, and D, which are standard for both extractors and coding theory.

that says that given a description of $\mathcal{S}$ the algorithm outputs all codewords in $\Gamma_A(\mathcal{S})$ in time poly(M, T). However, in many settings the oracle version is more natural as all we need is just to check membership in $\mathcal{S}$ and not to compute all the elements of $\mathcal{S}$.

We generally are interested in the following quantities, which are only well defined in an asymptotic sense.

DEFINITION 1.3. *Let $\mathcal{C} \subseteq [M]^T$ be a code. The* min-agreement min_agree$_\mathcal{C}$(S) *is the minimum* A *such that $\mathcal{C}$ can be list decoded for parameters* S *and* A. *If for a given* S, $NS(S, T)$ *is super-polynomial then we say* min_agree$_\mathcal{C}$(S) = ∞. *When the code is clear we may drop the subscript $\mathcal{C}$. The* extra-agreement *of $\mathcal{C}$,* extra_agree$_\mathcal{C}$, *is the minimum* Δ *such that for all* $S \in [MT]$, min_agree$_\mathcal{C}$(S) $\leq \frac{S}{M} + \Delta$.

1.2 Reed-Solomon and Random codes

Although list decoding was defined independently by Elias [3] and Wozencraft [19] in the 1950's, no non-trivial list decoding algorithm was known until the late 1980's. Since then, there have been several [4, 14, 6, 1]. For example, the work of Guruswami & Sudan [6], building upon Sudan [14], showed the following.

THEOREM 1. *[6] Let $\mathcal{RS} \subseteq [M]^T$ be a Reed-Solomon code,* $T \leq M$, $|\mathcal{RS}| = M^r$. *Then* min_agree$(S) \leq \sqrt{rS}$. *Furthermore, $\mathcal{RS}$ has an efficient list decoding for these parameters.*

Note that the bound is useless whenever $\sqrt{rS} > T$. In particular it is useless for the noise model above where for every i, $|\mathcal{S}_i| = \frac{M}{2}$ and $|\mathcal{S}| = \frac{MT}{2}$. Indeed, we show that Reed-Solomon codes are poor for large S.

THEOREM 2. *Let $\mathcal{RS} \subseteq [M]^T$ be a Reed-Solomon code,* $T \leq M$, $|\mathcal{RS}| = M^r$. *Then*

1. min_agree$_{\mathcal{RS}}(\frac{(M+1)T}{2}) = \infty$.

2. extra_agree$_{\mathcal{RS}} \geq T/2$.

On the other hand, we use the probabilistic method to prove the existence of good codes for combinatorial list decoding.

THEOREM 3. *Let* $M \geq 2$, $T \geq r$ *and* $S \in [MT]$. *There exist codes $\mathcal{C} \subseteq [M]^T$ of size* $C = M^r$ *such that:*

1. *For* $\frac{S}{M} \leq \frac{\log C}{8}$, min_agree$(S) \leq \log C = r \log M$.

2. extra_agree$_\mathcal{C} \leq 2\sqrt{T \log C}$.

The second case of this theorem applies, for example, to binary codes with $M = 2$ and $S = T$. The needed agreement is then $\frac{T}{2} + O(\sqrt{Tr})$ and it can be explicitly achieved by many binary codes (e.g., Hadamard or Reed-Solomon concatenated with Hadamard), see [15].

We focus on the first case, where $M = T$ (which is typical for Reed-Solomon codes) and $S = \frac{MT}{2}$ (i.e., the set $\mathcal{S}$ is large). In that case, the bound of Theorem 1 is useless, and the bound of Theorem 2 shows that no list decoding is possible with these parameters for Reed-Solomon codes. On the other hand Theorem 3 shows that there are codes that can list decode well even with these parameters, and furthermore the required agreement is only $\log C$.

A natural and important question is then to find explicit list decodable codes for larger alphabets that come close to these bounds, or at least that are better than Reed-Solomon codes.

1.3 Extractor codes

Before defining extractor codes, we discuss extractors. An extractor is a procedure to extract randomness from a defective random source, using a small additional number of truly random bits. Extractors were first defined and constructed in [10], and have since been improved by several authors. Extractors have found numerous applications to pseudo-randomness and explicit constructions [10, 18, 20, 16]. This paper gives yet another application.

We need a weaker variant of extractors called "slice extractors". The output of an extractor needs to be almost random with respect to any statistical test; the output of a slice extractor just needs to be almost random with respect to tests of a certain size. In fact, we just need the one-sided version of slice extractors. The definition we give is not quite analogous to the usual definition of extractors, but rather to an alternate definition. See Section 2 for more discussion on this point.

DEFINITION 1.4 (ONE-SIDED STRONG SLICE EXTRACTOR). *[11] $F : [C] \times [T] \to [M]$ is an (L, ϵ, p) one-sided strong slice extractor if for every $\mathcal{B} \subseteq [T] \times [M]$ of cardinality $\lceil p \cdot TM \rceil$,*

$$\left| \left\{ v \in [C] \quad : \quad \frac{|\Gamma(v) \cap \mathcal{B}|}{T} \geq \frac{|\mathcal{B}|}{TM} + \epsilon \right\} \right| \leq L$$

Our codes are simple to define. We associate a code to a one-sided slice extractor as follows:

DEFINITION 1.5. *For $F : [C] \times [T] \to [M]$, the corresponding extractor code $\mathcal{C}_F \subseteq [M]^T$ is defined as follows. For each $v \in [C]$ there is a codeword $w = w(v) = (w_1, \ldots, w_M) \in [M]^T$ defined by $w_i = F(v, i)$.*

We observe that one-sided slice extractors are equivalent to list decodable codes.

THEOREM 4. *Let $A \geq \frac{S}{M}$. Then $F : [C] \times [T] \to [M]$ is a $(L, \epsilon = \frac{A}{T} - \frac{S}{TM}, p = \frac{S}{TM})$ strong one-sided slice extractor iff $NS_{\mathcal{C}_F}(S, A) \leq L$. If $C \leq 2^T$ and if F is efficient then $\mathcal{C}_F$ has efficient encoding.*

Although a good extractor yields a good list decodable code with efficient encoding, we do not in general have an efficient decoding algorithm for such codes. We are able to show, however, that Trevisan's extractor [17] yields an efficient decoding algorithm:

THEOREM 5. *For every $M, C \leq 2^M$ and $\epsilon \geq \frac{1}{M}$ there is a code $\mathcal{C}_{TR} \subseteq [M]^T$ of size C such that*

- $T = 2^{O((\log \log C + \log \epsilon^{-1})^2)}$,

- extra_agree$(\mathcal{C}_{TR}) \leq T\epsilon$.

- *Efficient encoding and efficient probabilistic decoding.*

For the special case $M = T$ we can take ϵ so that $(\log \epsilon^{-1})^2 \approx \log M$, i.e., ϵ is about $2^{-\sqrt{\log M}} = 2^{-\sqrt{\log T}}$. In particular we can take any constant ϵ. We summarize this in Table 1.3.

We note that Trevisan [17] used list decodable binary codes to improve his extractor. However, he viewed it as less important than the use of pseudo-random generators, since he obtained good extractors without the use of codes. Our result is quite different, in that we use extractors to build codes over larger alphabets. Also, Sipser and Spielman [13] used expanders, which are related to extractors, to build codes. Again, our construction is quite different.

Code	min_agree($\frac{MT}{2}$)
$\mathcal{C}_{\mathcal{RS}}$	∞
$\mathcal{C}_{TR}$	$(\frac{1}{2} + \epsilon)T$
Non-explicit	$\frac{T}{2} + \sqrt{T \log C}$

Table 1: The case $M = T, S = \frac{MT}{2}$

1.4 Hardcore bits

The first application of list decoding [4] was obtaining hardcore bits, though it wasn't noticed as a list decoding algorithm at the time. For this use of list decoding, our notion of list decoding is more useful than the original notion.

The hardcore bits problem is: given a one-way function $f : \{0,1\}^r \to \{0,1\}^r$, can we obtain a hardcore function $h : \{0,1\}^r \to \{0,1\}^m$, such that $h(x)$ is hard to predict given $f(x)$? It is impossible to give one fixed h that works for all f, so Goldreich and Levin [4] allowed h to depend on some random bits y. We then want $h(x, y)$ that is unpredictable given $f(x)$ and y. Goldreich and Levin showed that the length of y could be taken to be r. Impagliazzo [8] gave a generic construction based on list decodable codes. This allowed him to get one hardcore bit while adding $O(\log r)$ auxiliary random bits. Using our codes, we can output many hardcore bits while adding $O(\log^2 r)$ auxiliary random bits.

1.5 Ramsey graphs

We conclude the paper with further evidence of the power of slice extractors. We show that slice extractors for certain parameters are equivalent to bipartite Ramsey graphs. While optimal extractors in the usual sense won't beat known constructions, optimal slice extractors will yield optimal bipartite Ramsey graphs.

2. EQUIVALENCE OF SLICE EXTRACTORS AND LIST DECODABLE CODES

We begin with the equivalence of slice extractors and list decodable codes, which is Theorem 4.

PROOF. $\mathsf{NS}_{\mathcal{C}_F}(S, A) \leq L$ iff for every set of points $\mathcal{S}$ of size at most S, the number of codewords with agreement A with $\mathcal{S}$ is at most L. This, however, happens iff $\forall \, \mathcal{S} \subseteq [T] \times [M]$ with $|\mathcal{S}| = S$ we have $|\{v \in [C] \; : \; |\Gamma(v) \cap \mathcal{S}| \geq A\}| \leq L$. We finish the proof by noticing that this is equivalent to the definition of F being a $(L, \epsilon = \frac{A}{T} - \frac{S}{TM}, p = \frac{S}{TM})$ one-sided strong slice extractor.

If F is efficient then $F(v, i)$, the ith bit of the vth codeword, can be computed in time polynomial in $\log C + \log T$. Since $\log C \leq T$ the running time is polynomial in T. $\square$

We remark that our definition of slice extractor is not analogous to the usual definition of extractors, but rather for an alternate version as discussed in [20]. A definition analogous to the usual extractor definition is the following, which we differentiate from the earlier definition by adding "in the original sense."

DEFINITION 2.1 (ONE-SIDED STRONG SLICE EXTRACTOR). [11] $F : [C] \times [T] \to [M]$ is an (L, ϵ, p) one-sided strong slice extractor in the original sense if for every $\mathcal{X} \subseteq [C]$ of cardinality more than L, and every $\mathcal{B} \subseteq [T] \times [M]$ of cardinality

$\lceil p \cdot TM \rceil$, if x is chosen uniformly from $\mathcal{X}$ and y is chosen uniformly from T, then

$$\Pr[F(x, y) \in \mathcal{B}] < \frac{|\mathcal{B}|}{TM} + \epsilon.$$

A one-sided strong slice extractor in the original sense is also a one-sided strong slice extractor, with the same parameters. Conversely, an (L, ϵ, p) one-sided strong slice extractor is an $(L/\epsilon, 2\epsilon, p)$ one-sided strong slice extractor in the original sense. See [20] for proofs of the analogous statements for extractors.

3. EXTRACTOR CODES WITH EFFICIENT DECODING

Our extractor codes with efficient decoding are based on Trevisan's extractor [17] and the improved version given in [12]. These extractors are based on weak designs and list decodable binary codes.

3.1 Weak Designs

DEFINITION 3.1 (WEAK DESIGN). [12] A family of sets $Z_1, Z_2, \ldots, Z_m \subseteq [t]$ is a weak (ℓ, ρ) design if

1. $\forall i \; |Z_i| = \ell$, and

2. $\forall i, \sum_{j < i} 2^{|Z_i \cap Z_j|} \leq \rho \cdot (m - 1)$.

We have:

LEMMA 3.1. [12] For every ℓ, m and $\rho > 1$, there exists a weak (ℓ, ρ) design $Z_1, Z_2, \ldots Z_m \subseteq [t]$ with $t = \left\lceil \frac{\ell}{\ln \rho} \right\rceil \cdot \ell$. Such a family can be found in time $\mathrm{poly}(m, t)$.

3.2 Binary codes

We need the following standard construction of list decodable binary codes (see [17]):

LEMMA 3.2. For every r and β, there is an efficient encoding $\mathcal{BC} = \mathcal{BC}_{r,\beta} : \{0,1\}^r \to \{0,1\}^{\bar{r}}$, where $\bar{r} = \mathrm{poly}(r, 1/\beta)$, such that every Hamming ball of radius $(1/2 - \beta)\bar{r}$ contains at most $1/\beta^2$ codewords.

3.3 Using Trevisan's extractor

Trevisan's extractor [17].

Parameters : $r, r \leq 2^m$, $\epsilon \geq \frac{1}{2^m}$, and $\rho = 2$. Set $\beta = \frac{\epsilon}{2m}$.

Binary code : Let $\mathcal{BC} = \mathcal{BC}_{r,\beta} : \{0,1\}^r \to \{0,1\}^{\bar{r}}$ denote the binary code given by Lemma 3.2, and denote $\hat{x} = \mathcal{BC}(x)$.

Weak Design : A weak (ℓ, ρ) design $Z_1, \ldots, Z_m \subseteq [t]$, with:

- $\ell = \log \bar{r} = O(\log r + \log \epsilon^{-1})$,
- $t = \ell \lceil \frac{\ell}{\ln \rho} \rceil = O((\log r + \log \epsilon^{-1})^2)$,

Input : $x \in \{0,1\}^r$.

Random coins : A random string $y \in \{0,1\}^t$.

Output : The output has m bits, $TR(x; y)_i = \hat{x}(y|Z_i)$.

Letting $T = 2^t$ and $M = 2^m$, note that $C_{TR} \subseteq [M]^T$ is of size 2^r. We now describe one step of the decoding procedure for C_{TR}.

Algorithm 3.3: One step in decoding C_{TR}.

Input : $\mathcal{S} \subseteq [T] \times [M]$, $|\mathcal{S}| = S$.

Random coins : Random strings $\alpha \in \{0,1\}^m$ and $\beta \in \{0,1\}^{t-\ell}$, where $M = 2^m$ and $T = 2^t$.

Algorithm : For every $1 \leq i \leq m$, and every set of possible truth tables $P_1, \ldots, P_{i-1}$, where P_j has size $2^{|Z_i \cap Z_j|}$, construct an output list as follows:

- For every $\gamma \in \{0,1\}^\ell$:
  - Define $b \in \{0,1\}^m$ by:
  $$b_j = \begin{cases} P_j(\gamma|_{Z_i \cap Z_j}) & \text{if } j < i \\ \alpha_j & \text{if } j \geq i \end{cases}$$
  - Denote $y = \gamma^{Z_i} \circ \beta^{[t] \setminus Z_i}$, i.e., γ in positions Z_i and β in positions $[t] \setminus Z_i$.
  - If $(y, b) \in \mathcal{S}$ we let $z_\gamma = b_i$ otherwise $z_\gamma = 1 - b_i$.
- We view $z : \{0,1\}^\ell \to \{0,1\}$ as $z \in \{0,1\}^{\bar{r}}$. For every codeword $\hat{v} \in \mathcal{BC}$ with agreement at least $(\frac{1}{2} + \frac{\epsilon}{2m})\bar{r}$ with z, we output the corresponding $v \in \{0,1\}^r$.

We claim:

LEMMA 3.3. *For every $\mathcal{S} \subseteq [T] \times [M]$ and $x \in \{0,1\}^r$ for which $|\Gamma(x) \cap \mathcal{S}| \geq A = \frac{S}{M} + T\epsilon$, the probability x appears in the output list of Algorithm 3.3 is at least $\frac{\epsilon}{2m}$.*

PROOF. The proof mostly follows Trevisan's proof [17], so we skip some details. Fix any $x \in \{0,1\}^r$ such that $|\Gamma(x) \cap \mathcal{S}| \geq A$. Define the test
$$1_\mathcal{S} : \{0,1\}^{t+m} \to \{0,1\}$$
which answers $1_\mathcal{S}(y, w) = 1$ iff $(y, w) \in \mathcal{S}$. Then
$$\Pr_{y \in \{0,1\}^t}(1_\mathcal{S}(y, TR(x, y)) = 1) \geq \frac{A}{T} \geq \frac{S}{TM} + \epsilon$$
$$\geq \Pr_{u \in \{0,1\}^{t+m}}(1_\mathcal{S}(u) = 1) + \epsilon$$

Define the hybrid distributions $D_0, \ldots, D_m$ where D_i picks y uniformly, then takes the first i bits from $TR(x, y)$ and the rest $\alpha_{i+1}, \ldots, \alpha_m$ chosen at random. Let us denote $\alpha_{[i+1, m]} = \alpha_{i+1}, \ldots, \alpha_m$. Then there is an i, $1 \leq i \leq m$, such that
$$\Pr_{y \in \{0,1\}^t, \alpha_{[i+1, m]}}(1_\mathcal{S}(D_i) = 1) \geq$$
$$Pr_{y \in \{0,1\}^t, \alpha_{[i, m]}}(1_\mathcal{S}(D_{i-1}) = 1) + \frac{\epsilon}{m}$$

We can split $y \in \{0,1\}^t$ into those bits in locations indexed by Z_i, which we denote by γ, and the rest of the bits in locations $[t] \setminus Z_i$ which we denote by β. Let us call the pair $(\beta, \alpha_{[i+1, m]})$ good if
$$\Pr_{\gamma \in \{0,1\}^\ell}(1_\mathcal{S}(D_i) = 1) \geq \Pr_{\gamma \in \{0,1\}^\ell, \alpha_i}(1_\mathcal{S}(D_{i-1}) = 1) + \frac{\epsilon}{2m}.$$

An averaging argument shows that
$$\Pr_{\beta \in \{0,1\}^{t-\ell}, \alpha_{[i+1, m]}}[(\beta, \alpha_{[i+1, m]}) \text{ is good}] \geq \frac{\epsilon}{2m}$$

For every good $(\beta, \alpha_{[i+1, m]})$ we check all truth tables $P_1, \ldots, P_{i-1}$, and, in particular, check the truth tables where $P_j(\gamma) = \hat{x}(\beta^{[t] \setminus Z_i} \circ \gamma^{Z_i})$ for all $j < i$. Yao's trick then implies that
$$\Pr_{\gamma \in \{0,1\}^\ell}[z_\gamma = \hat{x}_\gamma] \geq \frac{1}{2} + \frac{\epsilon}{2m}$$
and therefore x appears in the output. Thus, the algorithm is successful if $(\beta, \alpha_{[i+1, m]})$ are good, which occurs with probability at least $\frac{\epsilon}{2m}$. $\square$

When Algorithm 3.3 is repeated $2\frac{m}{\epsilon}r$ times, the probability x does not appear in the output list is at most e^{-r}. As the number of codewords is $C = 2^r$ the probability there is a codeword $v \in \mathcal{C}$ with $Ag(v, \mathcal{S}) \geq A$ that does not appear in the output list is at most $(2/e)^r$. The number of elements in the output list is at most $O(\frac{m}{\epsilon}r \cdot m \cdot 2^{\rho m} \cdot (\frac{m}{\epsilon})^2) = \frac{2^{(\rho + o(1))m}}{\epsilon^3}r$, because we repeat the basic decoding step $O(\frac{m}{\epsilon}r)$ times, each time we try the m possibilities for i, and the $2^{\rho m}$ possibilities for the tables $P_1, \ldots, P_{i-1}$, and then we get at most $O((\frac{m}{\epsilon})^2)$ possible answers (see Lemma 3.2). Taking $\rho = 2$, $r \leq M$ and $\epsilon \geq \frac{1}{M}$, we get a running time of poly(M), which completes the proof of Theorem 5.

4. HARDCORE BITS

Informally, the hardcore bits problem is: given a one-way function $f : \{0,1\}^r \to \{0,1\}^r$, can we obtain a hardcore function $h : \{0,1\}^r \to \{0,1\}^m$, such that $h(x)$ is hard to predict given $f(x)$? This is formalized as follows.

DEFINITION 4.1. *f is a one-way function with security (K, ϵ) if no probabilistic algorithm running in time K can invert f with probability at least ϵ, where the probability is over a random input and the coins of the algorithm.*

DEFINITION 4.2. *A test ϵ-distinguishes a distribution D if the probability that the test accepts under D differs from the probability under the uniform distribution by at least ϵ.*

DEFINITION 4.3. *$h : \{0,1\}^r \times \{0,1\}^t \to \{0,1\}^m$ is a hardcore function with security (K, ϵ) for a one-way function $f : \{0,1\}^r \to \{0,1\}^r$ if there is no probabilistic algorithm running in time K that, given $f(x)$ and y, can ϵ-distinguish $h(x, y) \circ y$, where the distinguishing probability is over random x, y, and the coins of the algorithm. h is hardcore with security (K, ϵ) if it is hardcore with security (K, ϵ) for all one-way functions with security $(K^2, \epsilon/2)$.*

Impagliazzo [8] gave a generic construction based on list decodable codes. Let $\mathcal{C} : \{0,1\}^r \to \{0,1\}^T$ be a list decodable binary code. Then $h(x, y) = \mathcal{C}(x)_y$ (the yth bit of the encoding of x) is a hardcore predicate. This shows that for $m = 1$ output bit the length of y could be much smaller than r. Impagliazzo's construction generalizes to the case of more output bits. In this case we have $\mathcal{C} : \{0,1\}^r \to [M]^T$, and the hardcore bits are $h(x, y) = \mathcal{C}(x)_y$.

THEOREM 6. *Let $\mathcal{C} : \{0,1\}^r \to [M]^T$ be a code with extra agreement at most $\frac{\epsilon}{2}T$, and the running time to find the close codewords is $o(K)$. Then $h(x, y) = \mathcal{C}(x)_y$ is hardcore with security (K, ϵ).*

PROOF. Suppose there is a function f and a statistical test that given $f(x)$ takes time K and ϵ-distinguishes $h(x,y) \circ y$, where $x \in \{0,1\}^r$ and $y \in [T]$ are chosen uniformly. Then this test has the form of an indicator function $1_{\mathcal{S}}$ for a set $\mathcal{S} \subseteq [M] \times [T]$ of size S. Assume without loss of generality that

$$\Pr_{x,y}[h(x,y) \circ y \in \mathcal{S}] \geq \frac{S}{MT} + \epsilon.$$

An averaging argument then gives that there is a set X' of size at least $\frac{\epsilon}{2} 2^r$, such that for all $x \in X'$,

$$\frac{1}{T} \mathsf{A}g(\mathcal{C}(x), S) = \Pr_y[h(x,y) \circ y \in \mathcal{S}] \qquad (1)$$

$$\geq \Pr_u[u \in \mathcal{S}] + \frac{\epsilon}{2} = \frac{S}{MT} + \frac{\epsilon}{2} \quad (2)$$

Since $\mathcal{C}$ has algorithmic extra agreement $\frac{\epsilon}{2}T$, there is an algorithm running in time $o(K) \cdot K$ which outputs a list of all x satisfying (1) (the decoding algorithm takes $o(K)$ time given the oracle, and each oracle call is an application of the statistical test which takes K time). To find an element of $f^{-1}(z)$, run this algorithm and apply f to each such x. This algorithm will be successful if some element of $f^{-1}(z)$ is in X', which happens with probability at least $\epsilon/2$. The total running time is at most K^2. $\square$

We can now use the extractor code $\mathcal{C}_{TR}$ to obtain a hardcore function outputting many bits, with auxiliary randomness only $O(\log^2(n/\epsilon))$.

THEOREM 7. Suppose $M = 2^m = o(K)$. Let $T = 2^t$ for $t = O(\log^2(n/\epsilon))$, and assume $T = o(K)$. Then the function $h(x,y) : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ defined by $h(x,y) = \mathcal{C}_{TR}(x)_y$ is hardcore with security (K, ϵ), and requires auxiliary randomness t.

5. REED-SOLOMON CODES

Theorem 2 follows from the following theorem by taking $b_2 = 2$.

THEOREM 8. Let $\mathcal{RS} \subseteq [M]^T$ be a Reed Solomon code with M^r elements, $T \leq M$ and M is a prime power. Suppose $M - 1 = b_1 b_2$ for some integers b_1 and b_2. Then,

$$\mathsf{N}S_{\mathcal{RS}}(T \cdot (b_1 + 1), T) \geq M^{\frac{r-1}{b_2}}.$$

The proof uses the multiplicative subgroups of F_M^*, an idea that was exploited in a more sophisticated way in [5].

PROOF. For $b_1 | M - 1$, F_M^* contains b_1 distinct solutions to the equation $x^{b_1} = 1$. Denote these solutions by $\{w_1, \ldots, w_{b_1}\}$, and define $w_0 = 0$. Let $\mathcal{S} = \{(i, w_j) : i \in [T], 0 \leq j \leq b_1\}$, $|\mathcal{S}| = T(b_1 + 1)$.

Let $\mathcal{B}$ be the set of polynomials p^{b_2} for some p of degree at most $\frac{r-1}{b_2}$. For every $q = p^{b_2} \in \mathcal{B}$ and i, either $p(i) = 0$ and then $q(i) = 0$ or $p(i) \neq 0$ and then $(q(i))^{b_1} = (p(i))^{b_1 b_2} = (p(i))^{M-1} = 1$, and so $q(i) \in \{w_1, \ldots, w_{b_1}\}$. Hence, if $\mathcal{RS}(q)$ denotes the codeword corresponding to q, $\mathsf{A}g(\mathcal{RS}(q), \mathcal{S}) = T$. Since distinct q of degree less than r give distinct codewords $\mathcal{RS}(q)$, $\mathsf{N}S_{\mathcal{RS}}(T(b_1 + 1), T) \geq |\mathcal{B}|$.

To lower bound $|\mathcal{B}|$, note that $F_M[X]$ is a unique factorization domain, which implies that for each $q \in \mathcal{B}$, the p such that $p^{b_2} = q$ is uniquely determined up to multiples

of F_M. Consequently, each $q \in \mathcal{B}$ can arise from at most M polynomials p of degree at most $\frac{r-1}{b_2}$ (in fact, at most b_2 such polynomials). Therefore $|\mathcal{B}| \geq M^{1 + \frac{r-1}{b_2}}/M$.

$\square$

6. BOUNDS FOR COMBINATORIAL LIST DECODING

In this section, using the probabilistic method we show non-constructively that codes with extremely good list decoding properties exist, and we also prove that our bounds are reasonably tight. In order to prove this theorem, we record a Chernoff bound [2] and some relevant notation.

6.1 Chernoff bound

DEFINITION 6.1. $R^+(n, \mu, \epsilon)$ is the smallest integer k such that $\Pr[\sum_{i=1}^n X_i \geq k] \leq \epsilon$ for $X_1, \ldots, X_n$ i.i.d. (independent and identically distributed) Boolean random variables with $\mathbb{E}(\sum_i X_i) = \mu$.

We will use Hoeffding's result [7] that the largest deviation for independent Boolean random variables occurs when they are i.i.d.

LEMMA 6.1. [7] For $X_1, \ldots, X_n$ independent Boolean random variables with $\mathbb{E}(\sum_i X_i) = \mu$, $\Pr[\sum_{i=1}^n X_i \geq R^+(n, \mu, \epsilon)] \leq \epsilon$.

The following implicitly gives an upper bound on R^+.

LEMMA 6.2. ([9], Theorem 4.1) Let $X_1, \ldots, X_n$ independent Boolean random variables, $X = \sum_i X_i$, $\mu = \mathbb{E}(X)$. Then, $\Pr[X \geq (1+\delta)\mu] \leq [e^\delta/(1+\delta)^{(1+\delta)}]^\mu$.

The special case $\mu \leq 1$ gives:

COROLLARY 6.1. Let $X_1, \ldots, X_n$ be independent Boolean random variables, $X = \sum_i X_i$, $\mu = \mathbb{E}(x)$. Assume $\mu \leq 1$. Then $\Pr[X \geq L] \leq (\frac{e}{L})^L$.

We use the following lemma from [9]:

LEMMA 6.3. For every n,

1. ([9] Exercise 4.1) If $\mu \leq \frac{\log \epsilon^{-1}}{2e}$ then $R^+(n, \mu, \epsilon) \leq \lceil \log \epsilon^{-1} \rceil$.

2. ([9] Theorem 4.3) If $\mu \geq \frac{\log \epsilon^{-1}}{2e}$ then $R^+(n, \mu, \epsilon) \leq \lceil 2e\mu \rceil$, and furthermore $R^+(n, \mu, \epsilon) \leq \lceil \mu + 2\sqrt{\mu \ln(\frac{1}{\epsilon})} \rceil$.

6.2 Bounds on list decodable codes

Using Lemma 6.3, Theorem 3 follows from the following theorem. We subtract 1 from R^+ in the lower bound because of integral issues.

THEOREM 9. Let $M \geq 2$, $T \geq r$ and $S \in [MT]$. Denote $\mu = \frac{S}{M}$.

- (upper bound) There exist codes $\mathcal{C} \subseteq [M]^T$ of size M^r such that $\mathsf{N}S_{\mathcal{C}}(S, R^+(T, \mu, M^{-r})) \leq 3 \max\left\{S, \sqrt{eTM}\right\}$.

- *(lower bound) For every code $\mathcal{C} \subseteq [\mathrm{M}]^{\mathrm{T}}$ of size M^r, and for every $t \leq r$, $\mathsf{N}S_{\mathcal{C}}(\mathrm{S}, R^+(\mathrm{T}, \mu, \mathrm{M}^{-t}) - 1) \geq \mathrm{M}^{r-t}$.*

PROOF. **Upper bound:**

Fix a set of pairs $\mathcal{S} = \{(a_i, b_i) \ : \ a_i \in [\mathrm{T}], b_i \in [\mathrm{M}]\}$ of size S. We chose $\mathcal{C}$ at random. We show that

- We expect that few elements of $\mathcal{C}$ have $R^+ = R^+(\mathrm{T}, \mu, \mathrm{M}^{-r})$ agreement with $\mathcal{S}$.

- The probability there are more than L elements of $\mathcal{C}$ with R^+ agreement with $\mathcal{S}$ is smaller than the reciprocal of the number of possible sets of points.

We conclude that almost all codes $\mathcal{C}$ are good for any set of points $\mathcal{S}$.

Formally, let $\mathcal{S}_i = \{(i, b_j) \in \mathcal{S}\}$ and $\mathrm{S}_i = |\mathcal{S}_i|$, i.e., $\mathrm{S}_1 + \ldots + \mathrm{S}_{\mathrm{T}} = \mathrm{S}$. Let Y_i be the random variable indicating that $(i, x_i) \in \mathcal{S}$, when $x = (x_1, \ldots, x_{\mathrm{T}}) \in [\mathrm{M}]^{\mathrm{T}}$ is chosen at random. Then, the $Y_1, \ldots, Y_{\mathrm{T}}$ are independent Boolean random variables and $\mathbb{E}(Y_i) = \frac{\mathrm{S}_i}{\mathrm{M}}$. Let us denote $Y = \sum_i Y_i$, $\mathbb{E}(Y) = \sum_i \frac{\mathrm{S}_i}{\mathrm{M}} = \frac{\mathrm{S}}{\mathrm{M}} = \mu$. We also note that $Y = \mathsf{A}g(x, \mathcal{S})$. Therefore, by Lemma 6.1,

$$\Pr_{x \in [\mathrm{M}]^{\mathrm{T}}}[\ \mathsf{A}g(x, \mathcal{S}) \geq R^+\ ] \ = \ \Pr[Y \ \geq R^+(\mathrm{T}, \mu, \mathrm{M}^{-r})]$$

$$\leq \ \mathrm{M}^{-r}$$

If $\mathsf{A}g(x, \mathcal{S}) \geq R^+$ we say x is bad for $\mathcal{S}$. The probability that a random x is bad for $\mathcal{S}$ is at most M^{-r}. $\mathcal{C}$ is a collection of M^r independent random strings, hence the expected number of bad strings in the code $\mathcal{C}$ is at most $\mathrm{M}^r \cdot \mathrm{M}^{-r} = 1$. Let $\mathrm{L} = 3 \max\left\{\mathrm{S}, \sqrt{e\mathrm{TM}}\right\}$. We say $\mathcal{S}$ is bad for $\mathcal{C}$ if $|\{w \in \mathcal{C} \ : \ \mathsf{A}g(w, \mathcal{S}) \geq R^+\}| \geq \mathrm{L}$. For $1 \leq \mathrm{S} \leq \mathrm{TM}$, let p_{S} denote the probability there is a set $\mathcal{S}$ of size S that is bad for $\mathcal{C}$. By Corollary 6.1

$$p_{\mathrm{S}} \leq \binom{\mathrm{TM}}{\mathrm{S}} \left(\frac{e}{L}\right)^{L}$$

For $\mathrm{S} \geq \sqrt{e\mathrm{TM}}$ this becomes:

$$p_{\mathrm{S}} \ \leq \ \binom{\mathrm{TM}}{\mathrm{S}}(\mathrm{S})^{-3\mathrm{S}}$$

$$\leq \ \left(\frac{e\mathrm{TM}}{\mathrm{S}}\right)^{\mathrm{S}} \cdot \mathrm{S}^{-3\mathrm{S}} \ \leq \ \mathrm{S}^{\mathrm{S}} \cdot \mathrm{S}^{-3\mathrm{S}} < \mathrm{S}^{-2\mathrm{S}}$$

and for $\mathrm{S} \leq \sqrt{e\mathrm{TM}}$ we have

$$p_{\mathrm{S}} \ \leq \ \binom{\mathrm{TM}}{\mathrm{S}}(\sqrt{e\mathrm{TM}})^{-3\sqrt{e\mathrm{TM}}}$$

$$\leq \ \left(\frac{e\mathrm{TM}}{\sqrt{e\mathrm{TM}}}\right)^{\sqrt{e\mathrm{TM}}}(\sqrt{e\mathrm{TM}})^{-3\sqrt{e\mathrm{TM}}}$$

$$< \ (\sqrt{e\mathrm{TM}})^{-2\sqrt{e\mathrm{TM}}}$$

In particular, the probability that there exists any bad set $\mathcal{S}$ is at most

$$\sum_{\mathrm{S}=1}^{\mathrm{TM}} p_{\mathrm{S}} \ \leq \ (\mathrm{TM})(\sqrt{e\mathrm{TM}})^{-2\sqrt{e\mathrm{TM}}} \ll 1.$$

Lower bound:

Our method is similar to that in [11]. Fix a code $\mathcal{C} \subseteq [\mathrm{M}]^{\mathrm{T}}$, say $\mathcal{C} = \{w^1, \ldots, w^{\mathrm{C}}\}$. We choose a random set of points $\mathcal{S} \subseteq [\mathrm{T}] \times [\mathrm{M}]$ of size S. Let us denote W_j the random variable that is 1 if $\mathsf{A}g(w^j, \mathcal{S}) \geq \mathrm{A} = R^+(\mathrm{T}, \mu, \mathrm{M}^{-t}) - 1$ and 0 otherwise. On the one hand we will show that $\mathbb{E}_{\mathcal{S}} W_j \geq \mathrm{M}^{-t}$ and therefore $\mathbb{E}_{\mathcal{S}} \sum_j W_j \geq \mathrm{CM}^{-t} = \mathrm{M}^{r-t}$. On the other hand, for every set $\mathcal{S}$ of cardinality S we have at most $\mathsf{N}S_{\mathcal{C}}(\mathrm{S}, \mathrm{A})$ codewords that have agreement A with $\mathcal{S}$ and therefore $\mathbb{E}_{\mathcal{S}} \sum_j W_j \leq \mathsf{N}S_{\mathcal{C}}(\mathrm{S}, \mathrm{A})$. Together, $\mathsf{N}S_{\mathcal{C}}(\mathrm{S}, \mathrm{A}) \geq \mathrm{M}^{r-t}$.

We still have to show that $\mathbb{E}_{\mathcal{S}} W_j \geq \mathrm{M}^{-t}$. Indeed, w^j defines a set $\Gamma^j = \{(i, w_i^j) \ : \ i = 1, \ldots, \mathrm{T}\}$. Let us denote by Y_i^j the random variable that is 1 if $(i, w_i^j) \in \mathcal{S}$ and 0 otherwise. Then $Y_1^j, \ldots, Y_{\mathrm{T}}^j$ are i.i.d. Boolean random variables. The random variable $Y^j = \sum_i Y_i^j$ is the agreement between w^j and $\mathcal{S}$, $Y^j = \mathsf{A}g(w^j, \mathcal{S})$, and $\mu = \mathbb{E}Y^j = \frac{\mathrm{S}}{\mathrm{M}}$. Therefore, using Lemma 6.1

$$\Pr_{\mathcal{S}}[Y^j \geq \mathrm{A}] \ = \ \Pr[Y^j \geq R^+(\mathrm{T}, \mu, \mathrm{M}^{-t}) - 1] \ \geq \ \mathrm{M}^{-t}$$

$\square$

7. SLICE EXTRACTORS AND RAMSEY GRAPHS

DEFINITION 7.1. *Let $G = (V_1 = [\mathrm{N}], V_2 = [\mathrm{N}], E)$ be a bipartite graph. We say G is b–Ramsey if for every two subsets $A_1 \subseteq V_1$, $A_2 \subseteq V_2$ of cardinality b, there is an edge and a non-edge between A_1 and A_2.*

Non-explicitly a random bipartite graph with degree $\mathrm{T} = \frac{\mathrm{N}}{2}$ is almost always $O(\log \mathrm{N})$–Ramsey. Explicitly, it is known how to build graphs G that are about $\sqrt{\mathrm{N}}$–Ramsey. The following is a well known generalization.

DEFINITION 7.2. *Let $G = (V_1 = [\mathrm{N}], V_2 = [\mathrm{N}], E)$ be a bipartite graph with degree T. We say G is a (b, ϵ)–Ramsey graph if for every two subsets $A_1 \subseteq V_1$, $A_2 \subseteq V_2$ of cardinality b, $\left|\frac{|E(A_1, A_2)|}{|A_1|\mathrm{T}} - \frac{|A_2|}{\mathrm{N}}\right| < \epsilon$.*

Indeed, if G is $(b, \epsilon = \frac{b}{\mathrm{N}})$–Ramsey and $b \leq \frac{\mathrm{N}}{2}$ then it is b–Ramsey. Being (b, ϵ)-Ramsey is equivalent to being a slice extractor. Before proving that, we first define slice extractors (as opposed to the one-sided strong version defined earlier). For this application, it is more useful to define slice extractors analogously to the original extractor definition.

DEFINITION 7.3 (SLICE EXTRACTOR). *[11] $F : [\mathrm{C}] \times [\mathrm{T}] \to [\mathrm{M}]$ is an $(\mathrm{L}, \epsilon, p)$ slice extractor if for every $\mathcal{X} \subseteq [\mathrm{C}]$ of cardinality at least L, and every $\mathcal{B} \subseteq [\mathrm{M}]$ of cardinality $\lceil p \cdot \mathrm{M} \rceil$, if x is chosen uniformly from $\mathcal{X}$ and y is chosen uniformly from T, then*

$$\left|\Pr[F(x, y) \in \mathcal{B}] - \frac{|\mathcal{B}|}{\mathrm{M}}\right| < \epsilon.$$

The following lemma is easy to check.

LEMMA 7.1. *G is (b, ϵ)–Ramsey iff it is a $(b, \epsilon, \rho = \frac{b}{\mathrm{N}})$ slice extractor.*

Non-explicitly, we show that slice extractors exist with degree which is a factor p smaller than the corresponding extractor.

LEMMA 7.2 (UPPER BOUND). *For every* $N, M \leq N, \epsilon > 0$ *and* $p \geq \epsilon$ *there exists an* $(L = pM, \epsilon, p)$ *slice extractor* $G = (V_1 = [N], V_2 = [M], E)$ *with degree* $T = O(\log(\frac{N}{L})\frac{p}{\epsilon^2})$.

PROOF. We take a random bipartite graph $G = (V_1 = [N], V_2 = [M], E)$ of degree T. If G is not a (L, ϵ, p) slice extractors then there are sets $A_1 \subseteq V_1$ of cardinality L, and $A_2 \subseteq V_2$ of cardinality pM such that each of the vertices of A_1 ϵ misses A_2. In particular half of the vertices of A_1 deviate by having too many or too few neighbors in A_2. In either case the probability for that event is at most $\exp\left(-\frac{\frac{L}{2}T\epsilon^2}{4p}\right)$ (See [9]. The lower deviation follows from Theorem 4.2, the upper deviation from Theorem 4.3 for $\frac{\epsilon}{p} \leq 2e - 1$). Thus, the probability the graph is not a (L, ϵ, p) slice extractor is at most

$$\binom{N}{L}\binom{M}{pM}\exp\left(-\frac{LT\epsilon^2}{8p}\right) \leq \left(\frac{Ne}{L}\right)^{2L}\exp\left(-\frac{LT\epsilon^2}{8p}\right),$$

which is less than one as long as $T > \frac{16(1+\ln(\frac{N}{L}))}{\epsilon^2}p$. $\square$

Thus, while a (L, ϵ) extractor requires degree $\Theta(\frac{\log N}{\epsilon^2})$, a (L, ϵ, p) slice extractor requires degree $\Theta(\frac{\log N}{\epsilon^2}p)$. The factor p gain does not come from the fact that there are fewer subsets A_1 and A_2 of cardinality pM then arbitrary subsets, but rather from the fact that when p is small the expectation of hitting A_2 is small and better Chernoff bounds apply.

This factor p is important. Indeed, if we try to use extractors for building Ramsey graphs we get a bottleneck at $\frac{b}{N} = \epsilon \leq \frac{1}{\sqrt{N}}$ and hence we can only get, roughly, $\sqrt{N}$–Ramsey graphs. Slice extractors have dependence only $T = \Omega(\frac{1}{\epsilon})$ and hence still work for very small b values. One can wonder whether the result can be improved. A matching lower bound was proved in [11]

LEMMA 7.3. *[11] [Lower bound] There exists a constant* $c > 0$, *such that if* $G = (V_1 = [N], V_2 = [M], E)$ *is a* (L, ϵ, p) *slice extractor of degree* T *with* $T \leq M/2, L \leq N/c, p \leq 1/10, pM \geq 1$ *and* $\epsilon \leq p/25$, *then*

$$T \geq \frac{p}{c\epsilon^2}\ln\frac{N}{cL}.$$

Acknowledgements

8. REFERENCES

[1] D. Boneh. Finding smooth integers in short intervals using CRT decoding. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 265–272, 2000.

[2] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–509, 1952.

[3] P. Elias. List decoding for noisy channels. In *1957-IRE WESCON Convention Record, Pt. 2*, pages 94–104, 1957.

[4] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 25–32, 1989.

[5] V. Guruswami, J. Hastad, M. Sudan, and D. Zuckerman. Combinatorial bounds for list decoding. In *Proceedings of the 38th Annual Allerton Conference on Communication, Control, and Computing*, pages 603–612, 2000.

[6] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45:1757–1767, 1999.

[7] W. Hoeffding. Probability inequalities for sums of bounded random variables. *American Statistical Association Journal*, 58:13–30, 1963.

[8] R. Impagliazzo, July 1997. Personal communication. For a written version, see [15].

[9] R. Motwani and Prabhakar Raghavan. *Randomized Algorithms*. MIT Press, 1995.

[10] N. Nisan and D. Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

[11] J. Radhakrishnan and A. Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, 2000.

[12] R. Raz, O. Reingold, and S. Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 149–158, 1999.

[13] M. Sipser and D. A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[14] M. Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13, 1997.

[15] M. Sudan. List decoding: Algorithms and applications. *SIGACT News*, 31:16–27, March 2000.

[16] M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the XOR lemma. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 537–546, 1999.

[17] L. Trevisan. Construction of extractors using pseudo-random generators. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 141–148, 1999.

[18] A. Wigderson and D. Zuckerman. Expanders that beat the eigenvalue bound: Explicit construction and applications. *Combinatorica*, 19(1):125–138, 1999.

[19] J.M. Wozencraft. List decoding. In *Quarterly Progress Report*, volume 48, pages 90–95. Research Laboratory of Electronics, MIT, 1958.

[20] D. Zuckerman. Randomness-optimal oblivious sampling. *Random Structures and Algorithms*, 11:345–367, 1997.